

REMARKS

The above Amendments and these Remarks are in reply to the Final Office Action mailed September 17, 2009.

Currently, claims 1-3, 5-13, 17-23, 28-30, 32-35, 37-42, 44-48 and 51-52 are pending. Applicants have amended claims 1-3, 5-6, 13, 17-18, 22-23, 33-35, 37, 39-42, 47 and 52. Claims 14-15, 24, 26-27, 49-50, 53-59 have been cancelled. Applicants respectfully request reconsideration of claims 1-3, 5-13, 17-23, 28-30, 32-35, 37-42, 44-48 and 51-52.

Rejection of Claims 13-32 and 59 Under 35 U.S.C. §112

Claims 13-32 and 59 have been rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 13 and 22 have been amended to remove the language that led to the rejection. Claim 59 has been cancelled. Therefore, the rejection to Claims 13-32 is believed to be moot.

It is respectfully requested that the rejection of claims 13-32 under 35 U.S.C. §112 be withdrawn.

Rejection of Claims 1-3, 5-8, 10, 12-15, 17-18, 20, 39-42, 44 and 47-58 Under 35 U.S.C. §103(a)

Claims 1-3, 5-8, 10, 12-15, 17-18, 20, 39-42, 44 and 47-58 have been rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,351,843 to Berkley, et al. (hereinafter “Berkley”) in view of U.S. Patent No. 6,738,965 to Webster (hereinafter “Webster”) and in further view of Call Graph Construction in Object-Oriented Languages by Grove, et al., (hereinafter “Grove”). The rejection to those claims not herein cancelled is respectfully traversed for the following reasons.

Currently amended claim 47 recites:

A process for monitoring, comprising:
 accessing a method;
 automatically determining whether said method is complex, said step of
 automatically determining includes automatically determining that said method is
 complex if said method satisfies the following criteria:

said method calls another method;
said method has an access level of public or package in the JAVA programming language; and
said method is not flagged by a compiler as being synthetic; and
adding a tracer to said method only if said method is automatically determined to be complex.

Applicants respectfully assert that the prior art fails to disclose, “adding a tracer to said method only if said method is automatically determined to be complex,” as claimed. As recited in claim 47, a method is complex if the method satisfies the criteria that: “said method calls another method,” “said method has an access level of public or package in the JAVA programming language,” **and** “said method is not flagged by a compiler as being synthetic.”

Applicants note that if the tracer were to be added to too many methods, then the performance of the software that contains the methods could be negatively impacted. However, if the tracer is not added to certain methods, then methods that should be traced might not be traced. It can be very challenging to automatically determine which method should be traced and which need not be traced. Applicants respectfully assert that the prior art provides no guidance that would suggest to one of ordinary skill in the art to add tracers to only the set of methods defined by the criteria recited in claim 47.

Neither Berkley nor Webster nor Grove, alone or in combination, disclose “adding a tracer to said method only if said method is automatically determined to be complex,” as claimed. Applicants first note that the Office Action appears to attack similar claim language presented in other claims individually. For example, the Office Action appears to assert that the prior art suggests that one could trace methods that call other methods. Next, the Office Action appears to assert that the prior art suggests that one could trace methods that are not synthetic. However, even if for the sake of argument the prior art made these individual suggestions, there still is no suggestion or motivation for tracing the combination of only methods that call other methods and methods that are not synthetic. For example, tracing such a combination would lead to excluding from tracing some methods that call other methods because they are synthetic.

Likewise, even if for the sake of argument the prior art suggested to trace methods that have an access level of public or package in the JAVA programming language there still is no suggestion or motivation for tracing the combination of **only** methods that call other methods and methods that access level of public or package in the JAVA programming language. For example, tracing such a combination would lead to excluding from tracing some methods that call other methods because they do not have an access level of public or package in the JAVA programming language. Likewise, there still is no suggestion or motivation for tracing the combination of **only** methods that are not synthetic and methods that access level of public or package in the JAVA programming language. For example, tracing such a combination would lead to excluding from tracing some methods that are not synthetic because they do not have an access level of public or package in the JAVA programming language.

Furthermore, there still is no suggestion or motivation for tracing the combination of **only** methods that call other methods, methods that are not synthetic, and methods that access level of public or package in the JAVA programming language.

Note that a problem solved by the limitations of claim 47 is the avoidance of tracing too many methods, while still tracing enough of the correct methods. It would not have been at all obvious to one of ordinary skill in the art that the claimed combination of criteria in claim 47 solves this problem.

Applicants also note the following reasons why one would not be motivated to modify Berkley to arrive at the limitations of currently amended claim 47. Berkley is concerned with inserting functions into an existing application executable of an object-oriented computer system without recompiling the executable. Berkley modifies runtime configuration settings to add a setting that specifies the function for at least one class of the application executable. Berkley runs the application executable using the modified configuration settings, and if it is determined that a function is active for a class then a re-direction stub is created dynamically to implement the function for the class (col. 2, lines 18-29).

Berkley discusses that in an object oriented system it can be difficult to determine the flow of the program. In this discussion, Berkeley refers to a call graph and the complexities that arise when

different methods call each other. Specifically, Berkley states that if an error occurred in a “current” method, then the execution of all “previous” methods is important to know the state that an object was in when the current method was called (col. 1, lines 43-62).

What the foregoing clearly indicates is that Berkley is simply not interested in determining which methods are synthetic, or which methods have an access level of either public or package in the JAVA programming language, as claimed. Nor would one of ordinary skill in the art have any reason to modify Berkley to determine which methods call another method, are not synthetic, and have an access level of either public or package in the JAVA programming language (and add a tracer to only those methods). To do so would not provide Berkley with the information such as, “the state on object was in when the current method was called.” Thus, Berkley would be rendered unsuitable for its intended purpose.

For all of the foregoing reasons, currently amended claim 47 is patentable over the cited prior art.

Currently amended claim 13 recites:

A process for monitoring, comprising:

determining which methods of a set of methods satisfy criteria for likely being at the automatically determining which methods of a set of methods call one or more other methods, and are synthetic; and

using a first tracing mechanism for said methods that call one or more other methods and are not synthetic without using said first tracing mechanism for methods that do not call one or more other methods or are synthetic.

Applicants respectfully assert that Berkley in view of Webster in further view of Grove fails to disclose, “using a first tracing mechanism for said methods that call one or more other methods and are not synthetic without using said first tracing mechanism for methods that do not call one or more other methods or are synthetic,” as claimed.

For reasons similar to those discussed in the response to claim 47, the prior art fails to teach or in any way suggest, “using a first tracing mechanism for said methods that call one or more other methods and are not synthetic without using said first tracing mechanism for methods that do not call one or more other methods or are synthetic.” For example, there is simply no teaching or suggestion

suggestion in the prior art for this claimed combination.

For all of the foregoing reasons, claim 13 is allowable.

Currently amended Claim 33 recites:

An apparatus capable of monitoring, comprising:
means for automatically determining whether a method calls another method;
means for automatically determining whether said method can be called by a sufficient scope of one or more other methods;
means for automatically determining whether said method is not a synthetic method; and
means for tracing said method for a particular purpose only if said method calls another method, said method can be called by a sufficient scope of one or more other methods, and said method is not a synthetic method.

For reasons similar to those discussed in the response to claim 47, the prior art fails to teach or in any way suggest, “means for tracing said method for a particular purpose only if said method calls another method, said method can be called by a sufficient scope of one or more other methods, and said method is not a synthetic method.” For example, there is simply no teaching or suggestion in the prior art for this claimed combination.

For all of the foregoing reasons, claim 33 is allowable.

Currently amended claim 1 recites:

A process for monitoring, comprising:
accessing a method;
automatically determining whether to modify said method, said step of automatically determining whether to modify said method includes automatically determining whether said method calls another method; and
modifying said method for a particular purpose only if said method calls another method.

Applicants respectfully assert that Berkley in view of Webster in further view of Grove fails to disclose, “modifying said method for a particular purpose only if said method calls another method,” as claimed.

For all of the foregoing reasons, claim 1 is allowable.

Currently amended claim 33 recites:

One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a process comprising:
 automatically determining whether to trace a method, said step of determining includes automatically determining whether said method calls another method; and
 tracing said method for a particular purpose only if said method calls another method.

For reasons similar to those discussed in the response to claim 1, claim 33 is respectfully believed to be allowable.

It is respectfully requested that the rejection of claims 1-3, 5-8, 10, 12-15, 17-18, 20, 39-42, 44 and 47-58 under 35 U.S.C. §103(a) be withdrawn.

Rejection of Claims 9, 11, 19, 21-24, 26-35, 37-38, 45-46 and 59 Under 35 U.S.C. §103(a)

Claims 9, 11, 19, 21-24, 26-35, 37-38, 45-46 and 59 have been rejected under 35 U.S.C. §103(a) as being unpatentable over Berkley in view of Webster and in further view of U.S. Patent No. 6,662,359 to Berry (hereinafter “Berry”). The rejection to those claims not herein cancelled is respectfully traversed for the following reasons.

Currently amended claim 22 recites:

One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a process comprising:
 automatically determining which methods of a set of methods to modify, said step of determining includes automatically determining which methods call one or more other methods and have an access level of either public or package in the JAVA programming language; and

modifying for a particular purpose only those methods that call one or more other methods and have an access level of either public or package in the JAVA programming language.

Applicants respectfully assert that Berkley in view of Webster in further view of Berry fails to disclose, “modifying for a particular purpose only those methods that call one or more other methods and have an access level of either public or package in the JAVA programming language,” as claimed.

For reasons similar to those discussed in the response to claim 47, the prior art fails to teach or in any way suggest, “modifying for a particular purpose only those methods that call one or more other methods and have an access level of either public or package in the JAVA programming language.” For example, there is simply no teaching or suggestion in the prior art for this claimed combination.

For all of the foregoing reasons, claim 22 is allowable.

It is therefore respectfully requested that the rejection of claims 9, 11, 19, 21-24, 26-35, 37-38, and 45-46 under 35 U.S.C. §103(a) be withdrawn.

Based on the above amendments and these remarks, reconsideration of claims 1-3, 5-13, 17-23, 28-30, 32-35, 37-42, 44-48 and 51-52 is respectfully requested.

The Examiner’s prompt attention to this matter is greatly appreciated. Should further questions remain, the Examiner is invited to contact the undersigned attorney by telephone.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 501826 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: December 17, 2009

By: /RonaldMPomerence/
Ronald M. Pomerence
Reg. No. 43,009

VIERRA MAGEN MARCUS & DENIRO LLP
575 Market Street, Suite 2500
San Francisco, California 94105-4206
Telephone: (415) 369-9660
Facsimile: (415) 369-9665